Contents

Intel® FPGA Temperature Sensor IP Core User Guide .......................................................... 3
   Intel FPGA Temperature Sensor Features ..................................................................... 3
   Intel FPGA Temperature Sensor Functional Description ............................................. 4
      Temperature Sensing Operation for Intel Arria 10 and Intel Cyclone 10 GX Devices ... 4
      Temperature Sensing Operation for Arria V, Arria V GZ, Stratix V, and Stratix IV Devices .......................................................... 5
   Using Clear Box Generator .......................................................................................... 7
   Intel FPGA Temperature Sensor Device Support ......................................................... 8
   Intel FPGA Temperature Sensor Parameters ............................................................... 9
   Transfer Function for Internal TSD .............................................................................. 10
   Intel FPGA Temperature Sensor IP Core Signals ....................................................... 10
   Intel FPGA Temperature Sensor Prototypes and Component Declarations ............... 12
      Verilog HDL Prototype ......................................................................................... 12
      VHDL Component Declaration ............................................................................. 12
   Document Revision History ....................................................................................... 13
The Intel® FPGA Temperature Sensor IP core configures the temperature sensing diode (TSD) block to utilize the temperature measurement feature in the FPGA.

**Related Information**

*Introduction to Intel FPGA IP Cores*

Provides general information about Intel FPGA IP cores.

### Intel FPGA Temperature Sensor Features

The following table lists the Intel FPGA Temperature Sensor IP core features.

**Table 1. Intel FPGA Temperature Sensor Features**

<table>
<thead>
<tr>
<th>Device</th>
<th>Features</th>
</tr>
</thead>
</table>
| Stratix® V, Stratix IV, Arria® V, and Arria V GZ | • An internal TSD with built-in 8-bit analog-to-digital converter (ADC) circuitry to monitor die temperature  
• A clock divider to reduce the frequency of the clock signal to 1 MHz or less before clocking the ADC  
• An asynchronous clear signal to reset the TSD block |
| Intel Arria 10 and Intel Cyclone® 10 GX | • An internal TSD with built-in 10-bit ADC circuitry clocked by 1 MHz internal oscillator to monitor die temperature  
• Does not require external clock source  
• An asynchronous clear signal to reset the TSD block |

**Note:** The Intel FPGA Temperature Sensor IP core does not have simulation model files and cannot be simulated.
Intel FPGA Temperature Sensor Functional Description

Temperature Sensing Operation for Intel Arria 10 and Intel Cyclone 10 GX Devices

Figure 1. Intel FPGA Temperature Sensor IP Core Top-Level Diagram for Intel Arria 10 and Intel Cyclone 10 GX Devices

The following lists the features for Intel FPGA Temperature Sensor IP core for Intel Arria 10 and Intel Cyclone 10 GX devices:

- For Intel Arria 10 and Intel Cyclone 10 GX devices, the Intel FPGA Temperature Sensor IP core supports the instantiation of temperature sensor block in your design from the IP Catalog.

- The Intel Arria 10 and Intel Cyclone 10 GX temperature sensor block runs at 1 MHz, where the clock signal is coming from the internal oscillator that is located in the temperature sensor block. Within the block, 10-bit ADC circuitry is included for converting sensor’s reading to digital output.

- The `corectl` signal is used as an enable signal. When asserting the `corectl` signal, the ADC starts the conversion and 10-bit data is available at `tempout` after 1,024 clock cycles. The `corectl` signal must remain asserted until the completion of 1,024 clock cycles. The `eoc` signal goes high for one clock cycle of the 1-MHz internal oscillator clock, indicating end of conversion. You can latch the data on `tempout` at the falling edge of `eoc`. When the `corectl` signal is left asserted, the ADC starts another conversion cycle and provides a new temperature value at `tempout`. However, if the `corectl` signal is de-asserted, `tempout` maintains its current temperature value until the `corectl` signal re-asserts, or the `reset` signal is asserted.

- You can reset the temperature sensor anytime by asserting the `reset` signal.

Related Information

Transfer Function for Internal TSD on page 10
Provides more information on how to calculate the temperature from the `tempout[9:0]` value.
Temperature Sensing Operation for Arria V, Arria V GZ, Stratix V, and Stratix IV Devices

Figure 2. Intel FPGA Temperature Sensor Block Diagram for Arria V, Arria V GZ, Stratix V, and Stratix IV Devices

This figure shows the top-level ports and the basic building blocks of the Intel FPGA Temperature Sensor IP core.

The Intel FPGA Temperature Sensor IP core runs at the frequency of the clk signal. The clk signal can run at a frequency of 80 MHz and below. The clock divider divides the clk signal to 1 MHz or less to feed the ADC. You can set the value of the clock divider using the Intel FPGA Temperature Sensor IP core parameter editor.

The ce signal connects to the output enable (oe) port of the clock divider block. Assert the ce signal to enable the Intel FPGA Temperature Sensor IP core. When you deassert the ce signal, the IP core disables the ADC, and maintains the previous values of the tsdcalo[7..0] and tsdcaldone signals unless you assert the clr signal, or reset the device. The clr signal is asynchronous, and you must assert the clr signal at least one clock cycle of the adcclk signal to clear the output ports.

Enabling the ADC allows you to measure the device temperature only once. To perform another temperature measurement, assert the clr signal, or reset the device. The clr signal is asynchronous, and you must assert the clr signal at least one clock cycle of the ADC clk signal to clear the output ports.

Note: When you choose not to create the ce port, the IP core connects the ce port to VCC. In this case, the ADC circuitry is always enabled. Intel recommends that you disable the ADC by deasserting the ce signal when the ADC is not in use to reduce power consumption.

During device power-up or when you assert the asynchronous clr signal, the Intel FPGA Temperature Sensor IP core sets the tsdcaldone port to 0 and the tsdcalo[7:0] signal to 11010101 or 0xB5. After 10 clock cycles of the adcclk signal, the Intel FPGA Temperature Sensor IP core asserts the tsdcaldone signal to indicate that the temperature sensing operation is complete and that the value of the tsdcalo[7:0] signal is valid. The value of the tsdcalo[7:0] signal corresponds to the device temperature range. For more information about the value of
tsdcalo[7:0] signals, refer to the Related Information. To start another temperature sensing operation, assert the clr signal for at least one clock cycle of the adcclk signal, or reset the device.

**Note:**
When you choose not to create the clr port, the Intel FPGA Temperature Sensor IP core connects the clr port to GND. In this case, you must reset the device to clear the output signals or start a temperature sensing operation. Intel recommends that you generate the clr port if you are planning to run the temperature sensing operation more than once.

If a derived PLL output clock is used to drive the Intel FPGA Temperature Sensor IP core, a minimum pulse violation might occur. When using the Intel FPGA Temperature Sensor IP core, you must ensure the clock applied must be less than or equal to 1 MHz. If you are using a higher frequency clock, the Intel FPGA Temperature Sensor IP core allows you use the 40 or 80 clock divider to reduce the clock frequency to be less than or equal to 1.0MHz.

**Related Information**

Intel FPGA Temperature Sensor IP Core Signals on page 10
Provides more information about the value of tsdcalo[7:0] that corresponds to the device temperature range.

**Generating the Intel FPGA Temperature Sensor IP Core Example Design**

To generate the Intel FPGA Temperature Sensor IP core, download the example design and follow these steps:

1. Open the alttemp_sense_ex1.zip file and extract alttemp_sense_ex1.qar.
2. In the Intel Quartus® Prime software, open the alttemp_sense_ex1.qar file and restore the archive file into your working directory.
3. On the IP Catalog window, search and click Intel FPGA Temperature Sensor.
4. In the New IP Instance dialog box, type tsd_s4 as your top-level file name.
5. In the Device family field, select Stratix IV.
6. Then, select your FPGA device family from the Device Family pull-down list. Click OK.
7. In the Parameter Editor, set the following parameter settings.

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the input frequency?</td>
<td>40 MHz</td>
</tr>
<tr>
<td>What is the clock divider value?</td>
<td>80 MHz</td>
</tr>
<tr>
<td>Create a clock enable port</td>
<td>Turned on</td>
</tr>
<tr>
<td>Create an asynchronous clear port</td>
<td>Turned on</td>
</tr>
</tbody>
</table>

8. Click Finish. The tsd_s4 module is built.

**Related Information**

Example Design
Compiling the Intel FPGA Temperature Sensor IP Core Example Design

To compile the Intel FPGA Temperature Sensor IP core in the Intel Quartus Prime software, follow these steps:

1. Open the top-level file `alttemp_sense_ex1.bdf` in the Intel Quartus Prime Block Editor software. This file contains the input and output assignments and a placeholder for the `tsd_s4` module.
2. To insert the `tsd_s4` module, double-click on the Block Editor window. The Symbol window appears.
3. Under Name, browse to the `tsd_s4.bsf` file.
4. Click OK.
5. Place the `tsd_s4` module onto the INSERT TSD_S4 BLOCK HERE placeholder so that the module aligns with the input and output ports.
6. On the Processing menu, click Start Compilation.
7. When the Full Compilation was successful message box appears, click OK.

Using Clear Box Generator

You can use clear box generator, a command-line executable, to configure parameters that are in the Intel FPGA Temperature Sensor IP core parameter editor. The clear box generator creates or modifies custom IP core variations, which you can instantiate in a design file. The clear box generator generates IP core variation file in Verilog HDL or VHDL format.

Note: Intel Arria 10 and Intel Cyclone 10 GX Intel FPGA Temperature Sensor IP core do not support clear box generation format.

To generate the Intel FPGA Temperature Sensor IP core using the clear box generator, perform the following steps:

1. Create a text file (.txt) that contains your clear box ports and parameter settings in your working directory.

   For example, `C:\altera\10.0\quartus\work\sample_param_test.txt`.

This figure shows a sample text file to generate the Intel FPGA Temperature Sensor IP core.
Figure 4. Sample Text File for Clear Box Generator

Note: Ensure that you enclose String-type values with double-quotes.

2. Access the command prompt of your operating system, and change the current directory to your working directory by typing the following command:

```
cd c:\altera\10.0\quartus\work\n```

The clear box executable file name is `clearbox.exe`.

Note: When you install the Intel Quartus Prime software, the `%QUARTUS_ROOTDIR%\bin` is added into your system's environment variables. Therefore, you can run the clear box command from any directory.

3. To view the available ports and parameters for this IP core, type the following command at the command prompt of your operating system:

```
clearbox alttemp_sense -h
```

4. To generate the Intel FPGA Temperature Sensor IP core variation file based on the ports and parameter settings in the text file, type the following command:

```
clearbox alttemp_sense -f *.txt
```

For example, `clearbox alttemp_sense -f sample_param_test.txt`

5. After the clear box generator generates the IP core variation files, you can instantiate the IP core module in a HDL file or a block diagram file in the Intel Quartus Prime software.

6. To view the estimated hardware resources that the Intel FPGA Temperature Sensor IP core uses, type the following command:

```
clearbox alttemp_sense -f sample_param_test.txt -resc_count
```

Note: This command does not generate a HDL file.

**Intel FPGA Temperature Sensor Device Support**

The Intel FPGA Temperature Sensor IP core supports the following device family:

- Intel Arria 10
- Intel Cyclone 10 GX
- Stratix V
Intel® FPGA Temperature Sensor Parameters

The parameters are applicable for all supported devices except Intel Arria 10 and Intel Cyclone 10 GX devices. There are no available parameters for Intel Arria 10 and Intel Cyclone 10 GX devices.

You can parameterize the Intel FPGA Temperature Sensor IP core using the IP Catalog and parameter editor, or with the command-line interface (CLI). Use the parameter editor to quickly specify parameters in a GUI.

Expert users may choose to instantiate and parameterize the IP core through the command-line interface using the clear box generator command. This method requires you to have command-line scripting knowledge.

This table lists the parameter editor and CLI parameter settings for the Intel FPGA Temperature Sensor IP core.

Table 3. Intel FPGA Temperature Sensor IP Core Parameter Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CLI Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the input frequency?</td>
<td>clk_frequency</td>
<td>Specifies the input frequency of the clk signal. The input frequency value is type string, and the value must be less than or equal to the clock divider value. The default value is 1.0</td>
</tr>
<tr>
<td>What is the clock divider value?</td>
<td>clock_divider_value</td>
<td>Specifies the clock divider value. The IP core divides the clock frequency value with the clock divider value before feeding the ADC. This option is only enabled when the clk signal frequency is more than 1 MHz. Intel recommends clocking the ADC with a 500 kHz signal. The CLI parameter is type integer. Ensure that you enable the clock divider by setting the clock_divider_enable parameter value to on. The default value is 40.</td>
</tr>
<tr>
<td>Create a clock enable port</td>
<td>ce</td>
<td>Specifies whether to turn on the asynchronous clock enable (ce) port. Turn on this option when you want to enable the Intel FPGA Temperature Sensor IP core. When you turn off this option, the clock enable port automatically connects to VCC.</td>
</tr>
<tr>
<td>Create an asynchronous clear port</td>
<td>clr</td>
<td>Specifies whether to turn on the asynchronous clear (clr) port.</td>
</tr>
<tr>
<td>Parameter</td>
<td>CLI Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Name</td>
<td>Legal Values</td>
<td>Name</td>
</tr>
</tbody>
</table>

Transfer Function for Internal TSD

The following figure shows the transfer function for internal TSD.

**Figure 5. ADC Transfer Function**

![ADC Transfer Function Diagram]

You can calculate the temperature from tempout[9:0] value using this formula:

\[
\text{Temperature} = \frac{(A \times C)}{1024} - B
\]

Where:
- A = 693
- B = 265
- C = decimal value of tempout[9..0]

Intel FPGA Temperature Sensor IP Core Signals

The following tables list the Intel FPGA Temperature Sensor IP core signals.
### Table 4. Intel FPGA Temperature Sensor IP Core Signals for Intel Arria 10 and Intel Cyclone 10 GX Devices

<table>
<thead>
<tr>
<th>Signals</th>
<th>Direction</th>
<th>Width (Bit)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>corectl</td>
<td>Input</td>
<td>1</td>
<td>Enables the temperature sensing feature by the IP core.</td>
</tr>
<tr>
<td>reset</td>
<td>Input</td>
<td>1</td>
<td>Resets the temperature sensing block.</td>
</tr>
<tr>
<td>tempout[9:0]</td>
<td>Output</td>
<td>10</td>
<td>Resets the temperature sensing 10-bit output data from internal ADC circuitry of temperature sensor block.</td>
</tr>
<tr>
<td>eoc</td>
<td>Output</td>
<td>1</td>
<td>Indicates end of internal ADC conversion. This signal goes high for one clock cycle of the 1-MHz internal oscillator clock and you can latch the data on tempout at the falling edge of eoc.</td>
</tr>
</tbody>
</table>

### Table 5. Intel FPGA Temperature Sensor IP Core Signals for Arria V, Arria V GZ, Stratix V, and Stratix IV Devices

<table>
<thead>
<tr>
<th>Signals</th>
<th>Direction</th>
<th>Width (Bit)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>Input</td>
<td>1</td>
<td>Input clock signal that runs at a frequency of 80 MHz and below. The internal clock divider reduces the frequency of the clk signal to 1 MHz or less before clocking the ADC.</td>
</tr>
<tr>
<td>ce</td>
<td>Input</td>
<td>1</td>
<td>The asynchronous clock enable signal for the clk signal. This signal turns on/off the Intel FPGA Temperature Sensor IP core that implements the TSD block. This is an active-high signal. By default, this port connects to VCC.</td>
</tr>
<tr>
<td>clr</td>
<td>Input</td>
<td>1</td>
<td>The asynchronous clear signal. When you assert the clr signal, the IP core sets the tsdcalo[7:0] signal to 11010101 (0xD5) and the tsdcaldone signal to 0. This is an active-high signal. By default, this port connects to GND.</td>
</tr>
<tr>
<td>tsdcalo[7:0]</td>
<td>Output</td>
<td>8</td>
<td>8-bit output signal that contains the analog-to-digital-conversion temperature value. The 8-bit value maps to a unique temperature value. During device power-up or when you assert the clr signal, the IP core sets the tsdcalo[7:0] to 11010101 (0xD5).</td>
</tr>
<tr>
<td>tsdcaldone</td>
<td>Output</td>
<td>1</td>
<td>This signal indicates the completion of the temperature sensing process. The IP core asserts this signal when the process is complete. During device power-up or when you assert the clr signal, the IP core sets the tsdcaldone to 0.</td>
</tr>
</tbody>
</table>

### Table 6. The Mapping of tsdcalo[7..0] Value to Arria V, Arria V GZ, Stratix V, and Stratix IV Devices Temperature

This table shows the value of tsdcalo[7:0] that corresponds to the device temperature range. The temperature specification ranges from -70°C to 127°C.

<table>
<thead>
<tr>
<th>Value of tsdcalo[7:0] in Hexadecimal</th>
<th>Temperature in Degree Celsius (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF</td>
<td>127</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>E4</td>
<td>100</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>D5</td>
<td>85</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

*continued...*
### Value of tsdcalo[7:0] in Hexadecimal

<table>
<thead>
<tr>
<th>Value</th>
<th>Temperature in Degree Celsius (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>80</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>B2</td>
<td>50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>9E</td>
<td>30</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>8A</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>76</td>
<td>-10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>6C</td>
<td>-20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>62</td>
<td>-30</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>4E</td>
<td>-50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3A</td>
<td>-70° C</td>
</tr>
</tbody>
</table>

**Related Information**

Transfer Function for Internal TSD on page 10

Provides more information on how to calculate the temperature from the `tempout[9:0]` value.

### Intel FPGA Temperature Sensor Prototypes and Component Declarations

#### Verilog HDL Prototype

The Verilog HDL prototype is located in the `<Intel Quartus Prime installation directory>\eda\synthesis\altera_mf.v`

#### VHDL Component Declaration

The VHDL component declaration is located in the `<Intel Quartus Prime installation directory>\libraries\vhdl\altera_mf\directory\altera_mf_components.vhd`. 
# Document Revision History

This table lists the changes made to the document.

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Intel Quartus Prime Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018.05.30</td>
<td>17.1</td>
<td>Updated the description of the <code>reset</code> signal in the <em>Intel FPGA Temperature Sensor IP Core Signals for Intel Arria 10 and Intel Cyclone 10 GX Devices</em> table.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date</th>
<th>Document Version</th>
<th>Changes Made</th>
</tr>
</thead>
</table>
| September 2017   | 2017.09.14       | • Beginning from the Intel Quartus Prime software version 17.1, the name of this IP core has been changed from Altera Temperature Sensor IP Core to Intel FPGA Temperature Sensor IP Core.  
• Added Intel Cyclone 10 GX devices information.  
• Changed instances of Quartus II to Intel Quartus Prime.  
• Added the link to Intel FPGA Temperature Sensor IP core example design.  
• Added the Transfer Function for Internal TSD section.  
• Updated the Intel FPGA Temperature Sensor IP Core Top-Level Diagram for Intel Arria 10 and Intel Cyclone 10 GX Devices figure.  
• Updated the Temperature Sensing Operation for Intel Arria 10 and Intel Cyclone 10 GX Devices section.  
• Updated the `eoc` description in the Intel FPGA Temperature Sensor IP Core Signals for Intel Arria 10 and Intel Cyclone 10 GX Devices table. |
| May 2017         | 2017.05.08       | Rebranded as Intel.                                                          |
| May 2015         | 2015.05.04       | • Added a link on how to calculate the temperature from the `tempout[9:0]` value.  
• Editorial updates. |
| December 2014    | 2014.12.15       | • Added Arria 10, Arria V, and Arria V GZ devices to the Device Support section.  
• Editorial changes to the warning message in Temperature Sensing Operation section.  
• Added Arria 10 devices information.  
• Updated template. |
| June 2013        | 3.1              | • Updated the "Temperature Sensing Operation" on page 3–1 to clarify that enabling the ADC allows you to measure the temperature of the device only once and to include a warning about a minimum pulse violation when input clock derived from a PLL.  
• Updated "Features" on page 1–1 to notify that this IP core does not provide simulation feature. |
| September 2010   | 3.0              | • Updated the Parameter Settings chapter.  
• Added the Prototypes and Component Declarations section.  
• Added the Clear Box Generator chapter. |
| February 2010    | 2.0              | Updated the Temperature Sensing Operation section.                          |
| November 2009    | 1.0              | Initial release.                                                            |